

information is quite helpful in, for example, knowing when a calibration or other service is due and to determine operator performance.

A significant item of the command table 540 is the generic command 542 data entity. Such a generic command data entity 542 (which may include command ID, command description, equipment type ID) permits the AMS controller 100 to simplify and streamline the commanding and controlling of a variety of equipment. By accessing the associated equipment table 520 via the equipment type ID, the AMS controller 100 can generate an equipment-specific command string from a generic command. Specifically, an equipment command string 546 is associated with the generic command 542 and to the equipment brand 522. A further mapping between the generic command 542 and command type 544 may also be used to provide an additional level of granularity to the command string sent to the equipment by the AMS controller 100. Thus, a generic script for conducting the stress testing processes that uses generic commands may be translated into equipment-specific command strings thereby greatly simplifying and streamlining the command process.

Table 500 may also be used to communicate with a variety of modules 15 under test. As mentioned above, some of the testing regimes may require sending commands to the modules 15 under test to, for example, place them in a particular mode or operational state. If a variety of modules 15 are being tested having different protocols, syntax, commands, etc then utilizing the table 500 would simplify sending commands to and receiving data from such modules 15.

As mentioned above, the invention also collects a variety of data from various equipment such as the test equipment 125. Such data may include the measurements made

by the test equipment 125 and data from the module under test 15 itself. Like the commands, the data received may be in a variety of formats and may use a variety of syntax. In order to translate such data into a consistent format for storage and analysis the parsing table 560 is used. After parsing the equipment-specific formatted data according to the parsing table 560, the data may be stored in the result table 430.

The result table 430 not only stores results but also associates those results to products via the product table 410 and processes via process table 450. As shown in Figure 14, the result table 430 may include the following data entities: result format data entity 432, result value data entity 434 and process-result value data entity 436. The result format data entity 432 stores result formatting and processing information such as result ID, result description, max decimal digits, sort order , etc. The result value 434 stores the actual test result value, a pass/fail result, test run ID, and result ID. The process result value data entity 436 maps the results of the process to the actual stress test process that led to those results and includes information items such as the process test run ID and result ID.

The result value 434 is also associated with the test criteria table 460 which stores various limits, thresholds and other criteria that can be used to determine whether the result value should be considered a pass or fail condition. In other words, the AMS controller 100 may utilized the test criteria table 460 and result value 434 to determine pass/fail of an associated product for a particular test process.

The association between a particular test result and the test process used to test the product is generally handled by the data association between the result table 430 and the process table 450. The process-result map 440 serves as an intermediary map for handing the many-to-many relationship between the result table 430 and the process table 450 and

may include both process ID and result ID keyed to similar information items in tables 430 and 450 for that purpose.

Figure 14 also shows exemplary data entities within the process table 450 which include process data entity 452, process test run data entity 454 and virtual oven data entity 456. The process data entity 452 identifies the test process by, for example, ID, name, description, etc. The virtual oven data entity 456 identifies the virtual oven 10 by, for example, ID, description, location, etc. The process test run data entity 454 associates the virtual oven item 456 with the processes 452 that may be executed in the virtual oven 10 and the process result value 436.

Furthermore, the process test run data entity 454 is associated with the process result value data entity 436. In other words, the process test run data entity 454 keeps track of a particular iteration (run) of a process and thereby permits segregation of different runs of the same process. By associating the process test run data entity 454 to the process result value data entity 436, the system can keep track of which process result values relate to which iteration of the process.

As further detailed in Figure 15, the test criteria table 460 may be expanded in order to handle different limit types, test runs and slots. More specifically, the test criteria table may include a limit type information item 462 (e.g. fixed limit range, percentage range, delta range), a run limit value information item 464 (e.g. storing actual limits against which the results are tested for pass/fail according to the type of limit), a test run information item 466 (to identify the test run process, the serial number of the product being tested, the slot ID (the module being tested may be plugged into a backplane that includes multiple slots for various PCB modules each of which may be subjected to one or more tests), and a slot